

EDN

COLUMNIST



DAVID BRUBAKER,
FUZZY-LOGIC
CONTRIBUTING EDITOR

Warning: Your current fuzzy systems are only half fuzzy

Ed note: The following letter is one of several the author recently received from readers.

I really enjoyed your column "Dismantling a fuzzy rule-based system shows its inner workings" (EDN, November 10, 1994, pg 189). Your column summarized fuzzy very well. I am one of the frustrated engineers who's really interested in fuzzy logic but hasn't yet had the chance to apply it on the job.

One concern regarding output membership functions: It seems like crisp outputs would be the same whether the output membership functions were symmetrical triangles or just singletons. If this [assumption] is true, why would we ever bother with triangles when singletons are easier to defuzzify?

Ray Johnson,
Lockheed Idaho Technologies

A kindred spirit! Ray is the only other person I know of who has realized that centroid defuzzification essentially is the same as using singletons. His letter also sparks a discussion I promised earlier on regarding defuzzification. Note that the shortcomings of fuzzy logic I point out are not weaknesses, but rather commentary on how fuzzy logic is currently being used in system design.

First, let's review what defuzzification is. Most fuzzy rule-based systems use membership functions to represent input conditions and output actions. These functions represent fuzzy sets or fuzzy values. Examples of fuzzy values include *low*, *fast*, or *heavy*. Examples of numeric (or crisp) values, with which we have had more experience, are 2, 5, or 23,462.789.

In a fuzzy system, when several rules fire simultaneously, the kernel typically defuzzifies and combines the specified output actions, expressed as fuzzy values, into a single, crisp output. (I use "typically" here because some systems do not defuzzify but rather maintain the output in its fuzzy form, often to pass on to a subsequent fuzzy stage.) Defuzzification is the operation that trans-

forms a fuzzy value into a crisp equivalent.

Currently, you can select from a number of techniques for defuzzification. The centroid, or center-of-gravity (COG), approach is the most popular. The kernel first lops off the top of each fuzzy output value at the truth value of the fired rule and then calculates the center of mass of each participating output. The center of these centers of mass indicates the single output action.

I know of four problems associated with centroid defuzzification. The first is computational intensity. Although calculating areas is not excessively difficult for triangular and trapezoidal functions, in real-time systems running on microcontrollers, the impact of doing so may be critical.

An example will best illustrate the second problem. Say a given output range has a high-end limit of 10. A membership function ranging, say, from 8 to 10 represents the maximum fuzzy value. Its COG is in the neighborhood of 9. Because centroid defuzzification calculates system outputs from COG figures, no system output greater than the COG of this function, that is, 9, will be possible—despite the desired maximum being 10. Of course, we can artificially overcome this shortcoming by extending the output domain beyond its actual limit and creating the membership function at the extreme, so that its COG coincides with the desired limit. However, this deviates from the close mapping between model and real world, which is a strength of a fuzzy system.

The third problem with centroid defuzzification is the point Ray Johnson makes in his letter. If output membership functions are vertically symmetric, the only difference on the resulting output value between them and singletons are the weights associated with different function areas. This situation results because the COG of a vertically symmetric function lies on a single vertical axis. Independent of whether the rule fired strongly or weakly (which occurs in response to changing input values), the designated output value remains constant.

Function areas, which singletons do not have, and which are used for weights when combining several actions, make up the fourth centroid problem. This issue tends to

PHOTOGRAPHY / LANE ENKELS

Military DC-DC Converters

Now
up to 500VDC
Output



• All Military Components

Semiconductors JAN-TX
Transformers MIL-T-27 (TF5S40ZZ)
Capacitors MIL-C-39003; MIL-C-55681
Resistors MIL-R-39008

• Standard Models

Single and Dual Outputs

• New MV Series

100 to 500VDC Output

• Low Profile

1.250" x .5" x .3" Height
up to 1.25 Watts, Hi Voltage
.35" Height

• Ambient Operating Temp.

-55°C to +85°C
(No heat sink or electrical
derating required)

• 4 Input Voltage Ranges

5, 12, 24 and 28VDC

• 100 Megohm

@ 500 VDC Isolation

@ 1000 VDC Hi Voltage

OPTIONS AVAILABLE per MIL-STD-883

- Stabilization Bake • Temperature Cycle
- Hi Temperature Burn-In 160 Hours at Full Power

Delivery—
stock to one week

See EEM or send direct
for Free PICO Catalog.
Call toll free 800-431-1064
In NY call 914-699-5514
FAX 914-699-5565

PICO Electronics, Inc.

453 North MacQuesten Pkwy., Mt. Vernon, N.Y. 10552
CIRCLE NO. 188

ing, that is, "I will accept any output in this admittedly broad range." And the narrow one's being narrow implies that we are quite certain as to its meaning, that is, "I want an output that is constrained to this specific region." "Wideness" corresponds to uncertainty and imprecision; "narrowness" corresponds to certainty and precision. If two rules fire with the same truth value, with centroid defuzzification, the wide (imprecise) output will take precedence over the narrow (precise) output. To me, this is counter-intuitive.

A better scheme would weight singletons based on the "area" of the region in the input space for which the given rule is active. About three years ago, I developed an inference method using this approach and called it "weighted singleton inference." I still use it, but no one else seems to have picked it up. I attribute this, at least in part, to a concurrent study I conducted comparing centroid, standard singletons, and weighted singletons in both closed- and open-loop control. For the most part, results were indistinguishable from the technique used.

While I'm on the soapbox, I have three more observations concerning defuzzification and centroids.

Fuzzy rule-based systems of today are only half fuzzy. A system's inputs are fuzzy because of the fuzzy input values, but, when using singleton output values or fuzzy output values with centroid defuzzification, outputs are not fuzzy. Any apparent fuzziness at the output comes from overlapping fuzzy input functions.

If a "half-fuzzy" system utilizes fuzziness only at its inputs, then, to be "fully fuzzy," a system must have both fuzzy inputs and fuzzy outputs. Small changes in the inputs must result in small changes in the output(s), even when only one rule is firing. In a fully fuzzy system, one would expect to see regions of constant output value only when using a crisp output function (a singleton), in addition to the fuzzy output functions.

Most fuzzy systems use fuzziness only as a means of interpolating outputs

a system response function obtained by dividing the input space into crisp, noncontiguous regions, assigning a constant output value to each of these regions, and letting the "fuzziness" provide smooth transitions (an interpolation) between the constant output values. Any interpolation system would work as well or better, and, for a two-input system, would be easier to implement.

The overall response of "half-fuzzy" systems is largely insensitive to the type of defuzzification used. So long as the transitions are smooth and interpolated (that is, they don't result in values outside the constant regions being connected) fuzzy systems are actually quite tolerant of various methods of transition calculation. Singletons, centroids, and weighted singletons all result in similar response functions, although centroids tend to have more "noise" when more than two rules are firing.

So, what does this all mean? First, we need a more powerful inference method (and, therefore, defuzzification technique), one that fully utilizes fuzziness at both input and output, and one that allows us to create "fully fuzzy" systems. This move is more difficult than it may seem, but it's evolving. Until its arrival, fuzzy-rule-based-system designers would do well to discard centroid as a defuzzification technique. The results are not worth the additional complexity. Use singletons instead; or, if you're a nonconformist, use weighted singletons. If you send me a self-addressed, stamped envelope, I'll send you a paper I wrote on them.

VOTE

Please use the Information Retrieval Service card to rate this article (circle one):

High Interest	Medium Interest	Low Interest
578	579	580

David Brubaker is a consultant in fuzzy-system design. You can reach him at Huntington Advanced Technology, 883 Santa Cruz Ave, Suite 31, Menlo Park, CA 94025-4608 or on the Internet at: brubaker@cup.portal.com.